IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: Charisius et al.	Group Art Unit: 2193
Application No: 09/839,526	Examiner: Wood, William H
Filed: 04/20/2001	Confirmation No. 2147

For: **Methods and Systems for Relating Data Structures and Object-Oriented Elements for Distributed Computing**

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT RESPONDING TO 28 NOVEMBER 2006 OFFICE ACTION

Sir:

In response to the 11/28/2006 Office Action, please consider the following amendments and arguments supporting allowance of the claimed subject matter. The applicant asserts that no new matter has been introduced by the amendments.

Amendments to the Claims are reflected in the listing of claims, which begins on page 2 of this paper.

Remarks/Arguments begin on page 35 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method in a data processing system, comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] includes computer instructions for performing the following computerized steps:

receiving an identification of a data structure with an attribute field in a database of data structures useable to form an object-oriented element from the data structure;

determining whether the data structure is associated with source code;

when it is determined that the data structure is associated with source code, determining whether the attribute field of the data structure is associated with an attribute in the source code; and

when it is determined that the attribute field is not associated with an attribute in the source code, generating a new attribute in the source code from the attribute field; and

operating the software development tool receiving user input to modify the source code.

2. (currently amended) The method of claim 1, further comprising the computerized steps of: when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure.

3. (currently amended) The method of claim 1, further comprising the computerized steps of:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code.

4. (currently amended) The method of claim 3, wherein the computerized step of removing the second attribute from the source code comprises the computerized step of removing a method associated with the second attribute from the source code.

5. (currently amended) The method of claim 3, further comprising the computerized step of displaying a graphical representation of the source code.

6. (currently amended) The method of claim 5, further comprising the computerized step of modifying the graphical representation of the source code to reflect the generation of the new attribute.

7. (currently amended) The method of claim 5, further comprising the <u>computerized</u> step of modifying the graphical representation of the source code to reflect the removal of the second attribute.

8. (currently amended) The method of claim 1, wherein the <u>computerized</u> step of determining whether the data structure is associated with the source code comprises the <u>computerized</u> step of searching a comment in the source code for the identification of the data structure.

9. (currently amended) The method of claim 1, wherein the <u>computerized</u> step of determining whether the data structure is associated with the source code comprises the <u>computerized</u> step of comparing a name for the source code with the identification of the data structure.

10. (currently amended) The method of claim 1, further comprising the <u>computerized</u> steps of: retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information.

11. (currently amended) The method of claim 10, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of retrieving the identification of the data structure and the access information from a configuration file.

12. (currently amended) The method of claim 10, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of retrieving the identification of the data structure and the access information from a comment of the source code.

13. (original) The method of claim 10, wherein the portion of the data structure comprises the attribute field of the data structure.

14. (original) The method of claim 1, wherein the source code comprises a class.

15. (original) The method of claim 1, wherein the source code comprises a distributed computing component.

16. (original) The method of claim 15, wherein the distributed computing component is an Enterprise JavaBean.TM.

17. (currently amended) The method of claim 1, wherein the <u>computerized</u> step of generating the new attribute in the source code comprises the <u>computerized</u> step of generating a method in the source code to access the attribute field of the data structure.

18. (currently amended) A method in a data processing system having source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device, the method comprising the steps of:

providing a software development tool <u>having a user interface</u> that is operable by a user to automatically reflect a modification [[of]] in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] <u>includes computer instructions</u> for performing the following <u>computerized</u> steps:

receiving an indication to update the source code; determining whether a first attribute in the source code is associated with a first attribute field in the data structure;

when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code;

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and when it is determined that a second attribute field is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field; and

~~operating the software development tool~~ <u>receiving user input</u> to modify the source code.

19. (currently amended) The method of claim 18, further comprising the <u>computerized</u> step of retrieving an identification of the data structure and access information for the secondary storage from a comment in the source code.

20. (currently amended) The method of claim 19, further comprising the <u>computerized</u> step of retrieving a portion of the data structure from the secondary storage device using the access information.

21. (original) The method of claim 20, wherein the portion comprises the first and the second attribute fields.

22. (currently amended) The method of claim 18, wherein the <u>computerized</u> step of removing the first attribute from the source code comprises the <u>computerized</u> step of removing a method associated with the first attribute from the source code.

23. (currently amended) The method of claim 18, wherein the <u>computerized</u> step of generating the second attribute in the source code comprises the <u>computerized</u> step of generating a method in the source code to access the second attribute field of the data structure.

24. (currently amended) A method in a data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of: providing a software development tool <u>having a user interface</u> that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool is for performing the following <u>computerized</u> steps:

receiving an indication that the data structure has been modified; and automatically reflecting the modification in the source code so as to avoid completely regenerating the source code; and ~~operating the software development tool~~ <u>receiving user input</u> to modify the source code.

25. (currently amended) The method of claim 24, wherein the <u>computerized</u> step of automatically reflecting the modification comprises the <u>computerized</u> steps of:

determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code.

26. (currently amended) The method of claim 25, wherein the <u>computerized</u> step of removing the first attribute from the source code comprises the <u>computerized</u> step of removing a first method associated with the first attribute in the source code.

27. (currently amended) The method of claim 25, wherein the <u>computerized</u> step of automatically reflecting the modification further comprises the <u>computerized</u> steps of:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field.

28. (currently amended) The method of claim 27, wherein the <u>computerized</u> step of generating the second attribute in the source code comprises the step of generating a second method in the source code to access the second attribute field of the data structure.

29. (currently amended) The method of claim 24, further comprising the <u>computerized</u> step of displaying a graphical representation of the source code.

30. (currently amended) The method of claim 29, further comprising the <u>computerized</u> step of modifying the graphical representation of the source code to reflect the modification.

31. (original) The method of claim 24, wherein the source code comprises a class.

32. (original) The method of claim 24, wherein the source code comprises a distributed computing component.

33. (original) The method of claim 32, wherein the distributed computing component is an Enterprise JavaBean.TM..

34. (currently amended) A method in a data processing system having a memory device with source code, the method comprising the steps of: providing a software development tool <u>having a user interface</u> that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] <u>includes computer instructions</u> for performing the following <u>computerized</u> steps:

determining whether the source code is associated with a data structure within a database of data structures useable to form an object-oriented element from the data structure;

when it is determined that the source code is associated with the data structure, determining whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating the first attribute field in the data structure;

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing the second attribute field from the data structure; and

~~operating the software development tool~~ <u>receiving user input</u> to modify the source code.

35. (currently amended) The method of claim 34, further comprising the <u>computerized</u> step of when it is determined that the source code is not associated with the data structure, generating the data structure from the source code.

36. (currently amended) The method of claim 34, wherein the <u>computerized</u> step of determining whether the source code is associated with the data structure comprises the <u>computerized</u> step of searching a comment in the source code for an identification of the data structure.

37. (currently amended) The method of claim 34, wherein the <u>computerized</u> step of determining whether the source code is associated with the data structure comprises the

computerized step of comparing a name for the source code with an identification of the data structure.

38. (currently amended) The method of claim 34, further comprising the computerized steps of: retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

39. (currently amended) The method of claim 38, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving an identification of the data structure and the access information from a configuration file.

40. (currently amended) The method of claim 38, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving an identification of the data structure and the access information from a comment of the source code.

41. (original) The method of claim 38, wherein the portion of the data structure comprises the first attribute field in the data structure.

42. (original) The method of claim 34, wherein the source code comprises a class.

43. (original) The method of claim 34, wherein the source code comprises a distributed computing component.

44. (original) The method of claim 34, wherein the first attribute field in the data structure is related to a method in the source code.

45. (currently amended) A method in a data processing system having a memory device with source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically modify source code, wherein the software development tool [[is]] includes computer instructions for performing the following computerized steps:

receiving an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code;

determining whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, removing the first attribute field from the data structure;

determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that the second attribute is not associated with the second attribute field in the data structure, adding the second attribute field to the data structure; and operating the software development tool receiving user input to modify the source code.

46. (currently amended) The method of claim 45, further comprising the computerized steps of:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

47. (currently amended) The method of claim 46, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving an identification of the data structure and the access information from a configuration file.

48. (currently amended) The method of claim 46, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving an identification of the data structure and the access information from a comment of the source code.

49. (original) The method of claim 46, wherein the portion of the data structure comprises the first attribute field in the data structure.

50. (original) The method of claim 45, wherein the source code comprises a class.

51. (original) The method of claim 45, wherein the source code comprises a distributed computing component.

52. (currently amended) A method in a data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool <u>having a user interface</u> that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] <u>includes computer instructions</u> for performing the following <u>computerized</u> steps:

receiving an indication that the source code has been modified; and automatically reflecting the modification in the data structure so as to avoid completely regenerating the data structure; and

~~operating the software development tool~~ <u>receiving user input</u> to modify the source code.

53. (currently amended) The method of claim 52, wherein the <u>computerized</u> step of automatically reflecting the modification comprises the <u>computerized</u> steps of: determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure.

54. (currently amended) The method of claim 52, further comprising the <u>computerized</u> steps of: determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure.

55. (currently amended) The method of claim 52, further comprising the <u>computerized</u> steps of: retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

56. (currently amended) The method of claim 55, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of retrieving an identification of the data structure and the access information from a configuration file.

57. (currently amended) The method of claim 55, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of retrieving an identification of the data structure and the access information from a comment of the source code.

58. (original) The method of claim 55, wherein the portion of the data structure comprises the first attribute field in the data structure.

59. (original) The method of claim 52, wherein the source code comprises a class.

60. (original) The method of claim 52, wherein the source code comprises a distributed computing component.

61. (original) The method of claim 52, wherein the first attribute field in the data structure is related to a method in the source code.

62. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] includes computer instructions for performing the following computerized steps:

receiving an identification of a data structure with an attribute field in a database of data structures useable to form an object-oriented element from the data structure; determining whether the data structure is associated with source code;

when it is determined that the data structure is associated with source code, determining whether the attribute field of the data structure is associated with an attribute in the source code; and

when it is determined that the attribute field is not associated with an attribute in the source code, generating a new attribute in the source code from the attribute field; and ~~operating the software development tool~~ receiving user input to modify the source code.

63. (currently amended) The computer-readable medium of claim 62, wherein the method further comprises the computerized steps of:

when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure.

64. (currently amended) The computer-readable medium of claim 62, wherein the method further comprises the <u>computerized</u> steps of:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code.

65. (currently amended) The computer-readable medium of claim 64, wherein the <u>computerized</u> step of removing the second attribute from the source code comprises the step of removing a method associated with the second attribute from the source code.

66. (currently amended) The computer-readable medium of claim 64, wherein the method further comprises the <u>computerized</u> step of displaying a graphical representation of the source code.

67. (currently amended) The computer-readable medium of claim 66, wherein the method further comprises the <u>computerized</u> step of modifying the graphical representation of the source code to reflect the generation of the new attribute.

68. (currently amended) The computer-readable medium of claim 66, wherein the method further comprises the <u>computerized</u> step of modifying the graphical representation of the source code to reflect the removal of the second attribute.

69. (currently amended) The computer-readable medium of claim 62, wherein the step of determining whether the data structure is associated with the source code comprises the computerized step of searching a comment in the source code for the identification of the data structure.

70. (currently amended) The computer-readable medium of claim 62, wherein the step of determining whether the data structure is associated with the source code comprises the computerized step of comparing a name for the source code with the identification of the data structure.

71. (currently amended) The computer-readable medium of claim 62, wherein the method further comprises the computerized steps of: retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information.

72. (currently amended) The computer-readable medium of claim 71, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving the identification of the data structure and the access information from a configuration file[[;]].

73. (currently amended) The computer-readable medium of claim 71, wherein the computerized step of retrieving the access information comprises the computerized step of

retrieving the identification of the data structure and the access information from a comment of the source code.

74. (original) The computer-readable medium of claim 71, wherein the portion of the data structure comprises the attribute field of the data structure.

75. (original) The computer-readable medium of claim 62, wherein the source code comprises a class.

76. (original) The computer-readable medium of claim 62, wherein the source code comprises a distributed computing component.

77. (original) The computer-readable medium of claim 76, wherein the distributed computing component is an Enterprise JavaBean.TM..

78. (currently amended) The computer-readable medium of claim 62, wherein the computerized step of generating the new attribute in the source code comprises the computerized step of generating a method in the source code to access the attribute field of the data structure.

79. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having source code that corresponds to a data structure within a database of data structures useable to

form an object-oriented element from the data structure on a secondary storage device, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool is for performing the following computerized steps:

receiving an indication to update the source code; determining whether a first attribute in the source code is associated with a first attribute field in the data structure, when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code; determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field; and operating the software development tool receiving user input to automatically reflect a modification in the source code.

80. (currently amended) The computer-readable medium of claim 79, wherein the method further comprises the computerized step of retrieving an identification of the data structure and access information for the secondary storage from a comment in the source code.

81. (currently amended) The computer-readable medium of claim 80, wherein the method further comprises the <u>computerized</u> step of retrieving a portion of the data structure from the secondary storage device using the access information.

82. (original) The computer-readable medium of claim 81, wherein the portion comprises the first and the second attribute fields.

83. (currently amended) The computer-readable medium of claim 79, wherein the <u>computerized</u> step of removing the first attribute from the source code comprises the <u>computerized</u> step of removing a method associated with the first attribute from the source code.

84. (currently amended) The computer-readable medium of claim 79, wherein the <u>computerized</u> step of generating the second attribute in the source code comprises the <u>computerized</u> step of generating a method in the source code to access the second attribute field of the data structure.

85. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:

provide a software development tool <u>having a user interface</u> that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the

source code, wherein the software development tool is for performing the following computerized steps: receiving an indication that the data structure has been modified; and

automatically reflecting the modification in the source code so as to avoid completely regenerating the source code; and

~~operating the software development tool~~ receiving user input to modify the source code.

86. (currently amended) The computer-readable medium of claim 85, wherein the step of automatically reflecting the modification comprises the computerized steps of:

determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code.

87. (currently amended) The computer-readable medium of claim 86, wherein the computerized step of removing the first attribute from the source code comprises the computerized step of removing a first method associated with the first attribute in the source code.

88. (currently amended) The computer-readable medium of claim 86, wherein the computerized step of automatically reflecting the modification further comprises the computerized steps of:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field.

89. (currently amended) The computer-readable medium of claim 88, wherein the computerized step of generating the second attribute in the source code comprises the computerized step of generating a second method in the source code to access the second attribute field of the data structure.

90. (currently amended) The computer-readable medium of claim 85, wherein the method further comprises the computerized step of displaying a graphical representation of the source code.

91. (currently amended) The computer-readable medium of claim 90, wherein the method further comprises the computerized step of modifying the graphical representation of the source code to reflect the modification.

92. (original) The computer-readable medium of claim 85, wherein the source code comprises a class.

93. (original) The computer-readable medium of claim 85, wherein the source code comprises a distributed computing component.

94. (original) The computer-readable medium of claim 93, wherein the distributed computing component is an Enterprise JavaBean.TM..

95. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a memory device with source code, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] includes computer instructions for performing the following computerized steps:

determining whether the source code is associated with a data structure within a database of data structures useable to form an object-oriented element from the data structure;

when it is determined that the source code is associated with the data structure, determining whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating the first attribute field in the data structure;

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing the second attribute field from the data structure; and

operating the software development tool receiving user input to modify the source code.

96. (currently amended) The computer-readable medium of claim 95, wherein the method further comprises the <u>computerized</u> step of when it is determined that the source code is not associated with the data structure, generating the data structure from the source code.

97. (currently amended) The computer-readable medium of claim 95, wherein the <u>computerized</u> step of determining whether the source code is associated with the data structure comprises the <u>computerized</u> step of searching a comment in the source code for an identification of the data structure.

98. (currently amended) The computer-readable medium of claim 95, wherein the <u>computerized</u> step of determining whether the source code is associated with the data structure comprises the <u>computerized</u> step of comparing a name for the source code with an identification of the data structure.

99. (currently amended) The computer-readable medium of claim 95, wherein the method further comprises the <u>computerized</u> steps of:

retrieving access information for a database that stores the data structure; and

retrieving a portion of the data structure from the database using the access information.

100. (currently amended) The computer-readable medium of claim 99, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of

retrieving an identification of the data structure and the access information from a configuration file.

101. (currently amended) The computer-readable medium of claim 99, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving an identification of the data structure and the access information from a comment of the source code.

102. (original) The computer-readable medium of claim 99, wherein the portion of the data structure comprises the first attribute field in the data structure.

103. (original) The computer-readable medium of claim 95, wherein the source code comprises a class.

104. (original) The computer-readable medium of claim 95, wherein the source code comprises a distributed computing component.

105. (original) The computer-readable medium of claim 95, wherein the first attribute field in the data structure is related to a method in the source code.

106. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a memory device with source code, the method comprising the steps of:

providing a software development tool <u>having a user interface</u> that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool [[is]] <u>includes computer instructions</u> for performing the following <u>computerized</u> steps:

receiving an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code;

determining whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, removing the first attribute field from the data structure; determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that the second attribute is not associated with the second attribute field in the data structure, adding the second attribute field to the data structure; and ~~operating the software development tool~~ <u>receiving user input</u> to modify the source code.

107. (currently amended) The computer-readable medium of claim 106, wherein the method further comprises the <u>computerized</u> steps of:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

108. (currently amended) The computer-readable medium of claim 107, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of

retrieving an identification of the data structure and the access information from a configuration file.

109. (currently amended) The computer-readable medium of claim 107, wherein the computerized step of retrieving the access information comprises the computerized step of retrieving an identification of the data structure and the access information from a comment of the source code.

110. (original) The computer-readable medium of claim 107, wherein the portion of the data structure comprises the first attribute field in the data structure.

111. (original) The computer-readable medium of claim 106, wherein the source code comprises a class.

112. (original) The computer-readable medium of claim 106, wherein the source code comprises a distributed computing component.

113. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a memory device with source code and a secondary storage device with a data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the

source code, wherein the software development tool [[is]] <u>includes computer instructions</u> for performing the following <u>computerized</u> steps:

receiving an indication that the source code has been modified; and automatically reflecting the modification in the data structure so as to avoid completely regenerating the data structure; and

~~operating the software development tool~~ <u>receiving user input</u> to modify the source code.

114. (currently amended) The computer-readable medium of claim 113, wherein the <u>computerized</u> step of automatically reflecting the modification comprises the <u>computerized</u> steps of:

determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and

when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure.

115. (currently amended) The computer-readable medium of claim 113, wherein the method further comprises the <u>computerized</u> steps of:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure.

29

116. (currently amended) The computer-readable medium of claim 113, wherein the method further comprises the <u>computerized</u> steps of:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

117. (currently amended) The computer-readable medium of claim 116, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of retrieving an identification of the data structure and the access information from a configuration file.

118. (currently amended) The computer-readable medium of claim 116, wherein the <u>computerized</u> step of retrieving the access information comprises the <u>computerized</u> step of retrieving an identification of the data structure and the access information from a comment of the source code.

119. (original) The computer-readable medium of claim 116, wherein the portion of the data structure comprises the first attribute field in the data structure.

120. (original) The computer-readable medium of claim 113, wherein the source code comprises a class.

121. (original) The computer-readable medium of claim 113, wherein the source code comprises a distributed computing component.

30

122. (original) The computer-readable medium of claim 113, wherein the first attribute field in the data structure is related to a method in the source code.

123. (currently amended) A data processing system comprising:

a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code;

a secondary storage device further comprising source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device;

a memory device further comprising a computer program that receives an indication to update the source code, that determines whether a first attribute in the source code is associated with a first attribute field in the data structure, and

when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, the program removes the first attribute from the source code, the program further determines whether a second attribute field in the data structure is associated with a second attribute in the source code, and

when it is determined that a second attribute field is not associated with a second attribute in the source code, the computer program generates the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field; and

a processor for running the computer program.

124. (currently amended) The data processing system of claim 123, wherein the computer program further retrieves an identification of the data structure and access information for the secondary storage from a comment in the source code.

125. (currently amended) The data processing system of claim 124, wherein the computer program further retrieves a portion of the data structure from the secondary storage device using the access information.

126. (original) The data processing system of claim 125, wherein the portion comprises the first and the second attribute fields.

127. (currently amended) The data processing system of claim 123, wherein when the computer program removes the first attribute from the source code, the computer program removes a method associated with the first attribute from the source code.

128. (currently amended) The data processing system of claim 123, wherein when the computer program generates the second attribute in the source code, the computer program generates a method in the source code to access the second attribute field of the data structure.

129. (currently amended) A data processing system comprising:

a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code; a secondary storage device further comprising source code;

a memory device further comprising a <u>computer</u> program that receives an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code, that determines whether a first attribute field of the data structure is associated with a first attribute in the source code, and

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, the <u>computer</u> program removes the first attribute field from the data structure, the <u>computer</u> program further determines whether a second attribute in the source code is associated with a second attribute field in the data structure, and

when it is determined that the second attribute is not associated with the second attribute field in the data structure, the <u>computer</u> program adds the second attribute field to the data structure; and a processor for running the <u>computer</u> program.

130. (currently amended) The data processing system of claim 129, wherein when the <u>computer</u> program further retrieves access information for a database that stores the data structure, and retrieves a portion of the data structure from the database using the access information.

131. (currently amended) The data processing system of claim 130, wherein when the <u>computer</u> program retrieves the access information, the <u>computer</u> program retrieves an identification of the data structure and the access information from a configuration file.

132. (currently amended) The data processing system of claim 130, wherein when the <u>computer</u> program retrieves the access information, the <u>computer</u> program retrieves an

33

identification of the data structure and the access information from a comment of the source code.

133. (original) The data processing system of claim 130, wherein the portion of the data structure comprises the first attribute field in the data structure.

134. (original) The data processing system of claim 129, wherein the source code comprises a class.

135. (original) The data processing system of claim 129, wherein the source code comprises a distributed computing component.

136. (currently amended) A system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the system comprising:

a software development tool having a user interface that is operable by a user to modify source code;

means wherein the software development tool includes computer instructions for receiving an indication that the data structure has been modified; and means computer instructions for automatically reflecting the modification in the source code so as to avoid completely regenerating the source code.

REMARKS

Rejection of Claims of Art Grounds in the 28 November 2006 Office Action, and Traversal Thereof

Claims 18-23 stand rejected under 35 U.S.C. 112, first paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter for which the applicant regards as the invention. Specifically, the Examiner holds that the word "of" in claim 18 should be removed (claims 19-23 are dependent on claim 18). Independent claim 18 is now amended to remove the word "of".

Claims 1-15, 17-32, 34-76, 78- 93 and 95-136 stand rejected under 35 U.S.C. 102(b) as being anticipated by Walton et al. (USPN 5,883,639). Claims 16, 33, 77 and 94 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Walton (USPN 5,883,639).

ARGUMENTS

The patent to Walton discloses a system for creating user interfaces for consumer products such VCR's, digital voltmeters, and the like. In particular, Walton discloses a visual software engineering system and method for developing visual prototypes on a computer display and connecting user code to the visual prototypes.

In contrast to Walton, the present invention is a software development tool for generating enterprise level source code in a much more efficient manner than is achievable with the prior art, including Walton. The present invention achieves improved efficiency by providing a method having computerized steps for automatically reflecting any modifications in a source code throughout a database to avoid completely regenerating the source code. The claims of the

present application are currently amended to clarify which steps of the present invention are computerized.

The Examiner relies on Figure 1 of Walton for his 35 U.S.C. 102(b) rejection pointing to the various graphical elements and components with attributes stored in a database. However, the flow diagram of the method steps shown in Figure 1 of Walton only show user method steps as opposed to the computerized method steps of the present application. The Figure 1 flow diagram of Walton along with its description in column 8, lines 44-65 predominantly discloses user method steps such as the following quote from column 8 line 45. "As illustrated, the interface designer picks graphical objects from libraries 100 of standard and user-created components and places these graphical objects in drawing and behavior editor 110." This quoted step is typical of the user steps disclosed by Walton in Figure 1. As such, the steps disclosed in Figure 1 of Walton are not equivalent to the computerized steps of the present invention.

As stated in MPEP §2131, a claim is anticipated under §102 only if each and every element as set forth in the claim, in as complete detail is found in a single prior art reference. While Walton does disclose various computerized steps in other Figures such as Figures 6-16, none of the disclosed steps perform the present invention's computerized steps for automatically reflecting any modifications in a source code to avoid completely regenerating the source code. Therefore, Walton cannot be properly held to anticipate the independent claims and their dependents as currently amended.

Moreover, the Federal Circuit has ruled on numerous occasions that a holding of "obviousness" requires some motivation, suggestion or teaching within the cited references that

would lead one skilled in the art to modify the cited reference or references as claimed by applicant. See, for example, In re Kotzab, 217 F3d 1365, 55 USPQ2d 1313 (Fed Cir. 2000):
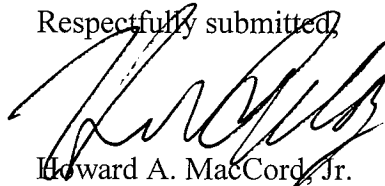
> "Most if not all inventions arise from a combination of old elements. See In re Rouffet, 149 F.3d 1350, 1357, 47 USPQ2d 1453, 1457 (Fed. Cir. 1998). Thus, every element of a claimed invention may often be found in the prior art. However, identification in the prior art of each individual part claimed is insufficient to defeat patentability of the whole claimed invention. Rather, to establish obviousness based on a combination of the elements disclosed in the prior art, there must be some motivation, suggestion or teaching of the desirability of making the specific combination that was made by the applicant. See In re Dance, 160 F.3d 1339, 1343, 48 USPQ2d 1635, 1637 (Fed. Cir. 1998); In re Gordon, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984). Even when obviousness is based on a single prior art reference, there must be a showing of a suggestion or motivation to modify the teachings of that reference. See B.F. Goodrich Co. v. Aircraft Breaking Sys. Corp., 72 F.3d 1577, 1582, 37 USPQ2d 1314, 1318 (Fed. Cir. 1996)."

The cited Walton patent and the present application both disclose software tools. However, the applicant's method, as specifically claimed, requires steps that are different from and neither suggested nor taught by the cited reference alone or in combination with other prior art. Moreover, there is no motivation to modify the methods of the cited Walton reference to incorporate the steps claimed by the applicant, since the result would be contrary to the aims of Walton, which are to develop visual prototypes of user interfaces for real world objects such as VCR's, digital voltmeters and the like. In contrast, the present invention is for developing enterprise level software such as that used by the banking industry, etc. Therefore, the independent claims and their dependents cannot be held as being obvious in view of the Walton reference.

CONCLUSION

In view of the foregoing amendments and for the above reasons, it is believed that this application is now in condition for allowance. If unresolved issues remain, the Examiner is invited to telephone applicant's attorney at the number below.

Respectfully submitted,

Howard A. MacCord, Jr.
Registration No. 28,639
MacCord Mason PLLC

Date: February 28, 2007
File No.: 7399-015